Balanced Allocations

(Extended abstract)

[To appear in the proceedings of STOC 1994.]

Y. Azar* A. Z. Broder† A. R. Karlin† E. Upfal‡

Abstract

Suppose that we sequentially place n balls into n boxes by putting each ball into a randomly chosen box. It is well known that when we are done, the fullest box has with high probability $\ln n/\ln\ln n(1+o(1))$ balls in it. Suppose instead, that for each ball we choose two boxes at random and place the ball into the one which is less full at the time of placement. We show that with high probability, the fullest box contains only $\ln\ln n/\ln 2+O(1)$ balls – exponentially less than before. Furthermore, we show that a similar gap exists in the infinite process, where at each step one ball, chosen uniformly at random, is deleted, and one ball is added in the manner above. We discuss consequences of this and related theorems for dynamic resource allocation, hashing, and on-line load balancing.

1 Introduction

Suppose that we sequentially place n balls into n boxes by putting each ball into a randomly chosen box. Properties of this random allocation process have been extensively studied in the probability and statistics literature. (See e.g. [15, 12].) One of the classical results in this area is that when the process has terminated, the fullest box has, with high probability, $\ln n/\ln\ln n(1+o(1))$ balls in it[1].

Consider a variant of the process above whereby each ball comes with d possible destinations, chosen independently and uniformly at random. The ball is placed in the least full box among the d possible locations. Surprisingly, even for $d=2$, when the process terminates the fullest box has only $\ln\ln n/\ln 2+O(1)$ balls in it. Thus, this apparently minor change in the random allocation process results in an exponential decrease in the maximum occupancy per location. The analysis of this process is summarized as follows

Theorem 1 *Suppose that m balls are sequentially placed into n boxes. Each ball is placed in the least full box, at the time of the placement, among d boxes, $d \geq 2$, chosen independently and uniformly at random. Then after all the balls are placed*

- *With high probability, as $n\to\infty$ and $m \geq n$, the number of balls in the fullest box is $\ln\ln n/\ln d(1+o(1))+\Theta(m/n)$.*
- *In particular, with high probability, as $n\to\infty$ and $m=n$, the number of balls in the fullest box is $\ln\ln n/\ln d+\Theta(1)$.*
- *Any other on-line placement strategy results in stochastically more balls in the fullest box.*

It is also interesting to study the infinite version of the random allocation process. There, at each step a ball is chosen uniformly at random and removed from the system, and a new ball appears. The new ball comes with d possible destinations, chosen independently at random, and it is placed into the least full box among these d possible destinations.

The analysis of the case $d=1$ in this infinite stochastic process is simple since the location of any ball does not depend on the locations of other balls in the system. Thus, for $d=1$, in the stationary distribution, with high probability the fullest box has

*Tel Aviv University, Israel. E-mail: azar@cs.stanford.edu.

†Digital Systems Research Center, 130 Lytton Avenue, Palo Alto, CA 94301, USA. E-mail: broder@src.dec.com, karlin@src.dec.com.

‡IBM Almaden Research Center, San Jose, CA 95120, USA, and Department of Applied Mathematics, The Weizmann Institute of Science, Rehovot, Israel. Work at the Weizmann Institute supported in part by the Norman D. Cohen Professorial Chair of Computer Science. E-mail: eli@wisdom.weizmann.ac.il.

[1] G. Gonnet [11] has proven a more accurate result, $\Gamma^{-1}(n)-3/2+o(1)$.

$\Theta(\log n/\log\log n)$ balls. The analysis of the case $d \geq 2$ is significantly harder, since the locations of the current n balls might depend on the locations of balls that are no longer in the system. We prove that when $d \geq 2$, in the stationary distribution, the fullest box has $\ln\ln n/\ln d + O(1)$ balls, with high probability. Thus, the same exponential gap holds in the infinite process. Theorem 2 is proven in section 4.

Theorem 2 *Consider the infinite process with $d \geq 2$, starting in an arbitrary state. For any fixed $T > n^3$, the fullest box at time T contains, with high probability, less than $\ln\ln n/\ln d + O(1)$ balls.*

1.1 Applications.

Our results have a number of interesting applications to computing problems. We elaborate here on three of them:

Dynamic Resource Allocation: Consider a scenario in which a user or a process has to choose on-line between a number of identical resources (choosing a server to use among the servers in a network; choosing a disk to store a directory; etc.). To find the least loaded resource, users may check the load on all resources before placing their requests. This process is expensive, since it requires sending an interrupt to each of the resources. A second approach is to send the task to a random resource. This approach has minimum overhead, but if all users follow it, the difference in load between different servers will vary by up to a logarithmic factor. Our analysis suggests a more efficient solution. If each user samples the load of two resources and sends his request to the least loaded, the total overhead is small, and the load on the n resources varies by only a $O(\log\log n)$ factor.

Hashing: The efficiency of a hashing technique is measured by two parameters: the expected and the maximum access time. Our approach suggests a simple hashing technique, similar to hashing with chaining. We call it *2-way chaining*. It has $O(1)$ expected, and $O(\log\log n)$ maximum access time. We use two random hash functions. The two hash functions define two possible entries in the table for each key. The key is inserted to the least full location, at the time of the insertion. Keys in each entry of the table are stored in a linked list. Assume that n keys are sequentially inserted by this process to a table of size n. The expected insertion and look-up time is $O(1)$, and our analysis proves that with high probability the maximum access time is $\ln n \ln n/\ln 2 + O(1)$, versus to the $\Theta(\log n/\log\log n)$ time when one random hash function is used.

An advantage of our scheme over some other known techniques for reducing worst case behavior of hashing (e.g. [10, 9, 8]) is that it uses only two hash functions, it is easy to parallelize, and does not involve re-hashing of data. Most of the other commonly used schemes partition the available memory into multiple tables, and use a different hash function in each sub-table. For example, the Fredman, Komlos, Szemeredi scheme for perfect hashing [10], uses up to n different hash functions to get $O(1)$ worst case access time (not on-line however), and the algorithm of Broder and Karlin [8] uses $O(\log\log n)$ hash functions to achieve $O(\log\log n)$ maximum access time, on-line, but using re-hashings.

A recent result of Karp, Luby, and Meyer auf der Heide [13] also exhibits a dramatic improvement when switching from one hash function to two in the context of PRAM simulations. In fact it is possible to use a result from [13] to derive a weak form of our Theorem 4 as explained in Appendix A.

Competitive On-line Load Balancing: Consider the following on-line load balancing problem: We are given a set of n servers and a sequence of arrivals and departures of tasks. Each task comes with a list of servers on which it can be executed. The load balancing algorithm has to assign each task to a server on-line, with no information on future arrivals and departures of tasks. The goal of the algorithm is to minimize the maximum load on any server. The quality of an on-line algorithm is measured by the *competitive ratio*: the ratio between the maximum load it achieves and the maximum load achieved by the optimal off-line algorithm that knows the whole sequence in advance. This load balancing problem models for example, communication in heterogeneous networks containing workstations, I/O devices, etc. Servers correspond to communication channels and tasks to requests for communication links between devices. A network controller must coordinate the channels so that no channel is too heavily loaded.

On-line load balancing has been studied extensively against worst-case adversaries [7, 5, 2, 3, 6, 4]. For permanent tasks (tasks that arrive but never depart), Azar, Naor and Rom [7] showed that the competitive ratio of the greedy algorithm is $\log n$ and that no algorithm can do better. For temporary tasks, the works of Azar, Broder and Karlin [5] and Azar, Kalyanasundaram, Plotkin, Pruhs and Waarts [6] show that there is an algorithm with competitive ratio $\Theta(\sqrt{n})$ and that no algorithm can do better.

It is interesting to compare these high competitive ratios, obtained from inputs generated by an adversary, to the competitive ratio against randomly generated inputs. Our results show that under reasonable probabilistic assumptions the competitive ratios for both permanent and temporary tasks are significantly better. In the case of permanent tasks, if the set of servers on which a task can be executed is chosen at random, the competitive ratio decreases from $\Theta(\log n)$ to $\Theta(\log\log n)$. In the case of temporary tasks, if we further assume that at each time step a randomly chosen existent task is replaced by a new task, then at any fixed time the ratio between the maximum online load and the maximum offline load is $\Theta(\log\log n)$ with high probability. Further details are presented in Section 5.

2 Definitions and Notation

We consider two stochastic processes: the finite process and the infinite process.

The Finite Process: There are n boxes, initially empty, and m balls. Each ball is allowed to go into $d \geq 1$ boxes chosen independently and uniformly at random. The balls arrive one by one, and a *placement algorithm* must decide on-line (that is, without knowing what choices are available to future balls) in which box to put each ball as it comes. Decisions are irrevocable. We will subsequently refer to this setup as a (m, n, d)-problem.

The Infinite Process: There are n boxes, initially containing n balls in an arbitrary state. (For example, all the balls could be in one box.) At each step, one random ball is removed, and one new ball is added; the new ball is allowed to go into $d \geq 1$ boxes chosen independently and uniformly at random. Once again, a placement algorithm must decide on-line (that is, without knowing what choices are available to future balls and without knowing which ball will be removed at any future time) in which box to put each arriving ball. Decisions are irrevocable.

We use the following notations for the random variables associated with a placement algorithm A. Note that the state at time t refers to the state immediately after the placement of the t'th ball.

$\lambda_j^A(t)$ called the *load* of box j, is the number of balls in box j at time t, resulting from algorithm A acting on a random input.

$\nu_k^A(t)$ is the number of boxes that have load k at time t.

$\nu_{\geq k}^A(t)$ is the number of boxes that have load $\geq k$ at time t, that is $\nu_{\geq k}^A(t) = \sum_{i\geq k} \nu_i^A(t)$.

h_t^A called the *height* of ball t (= the ball that arrives at time t), is the number of balls at time t in the box where ball t is placed.

$\mu_k^A(t)$ is the number of balls that have height k at time t.

$\mu_{\geq k}^A(t)$ is the number of balls that have height $\geq k$ at time t, i.e. $\mu_{\geq k}^A(t) = \sum_{i\geq k} \mu_i^A(t)$.

We omit the superscript A when it is clear which algorithm we are considering. Constants were chosen for convenience, and we made no attempts to optimize them.

Algorithm GREEDY assigns ball j to the box the has the lowest load among the d random choices that j has. We use the superscript G for GREEDY.

The basic intuition behind the proofs that follow is simple: Let $p_i = \mu_{\geq i}/n$. Since the available choices for each ball are independent, and $\nu_{\geq i} \leq \mu_{\geq i}$, we roughly have ("on average" and disregarding conditioning) $p_{i+1} \leq p_i^d$, which implies a doubly exponential decrease in p_i, once $\mu_{\geq i} < n/2$. Of course the truth is that $\mu_{\geq i+1}$ is strongly dependent on $\mu_{\geq i}$ and a rather complex machinery is required to construct a correct proof.

3 The Finite Process

We use the notation $B(n,p)$ to denote a binomially distributed random variable with parameters n and p, and start with the following standard lemma, whose proof is omitted.

Lemma 3 *Let $X_1, X_2, \ldots, X_n$ be a sequence of random variables with values in an arbitrary domain, and let $Y_1, Y_2, \ldots, Y_n$ be a sequence of binary random variables, with the property that $Y_i = Y_i(X_1, \ldots, X_i)$. If*

$$\mathbf{Pr}(Y_i = 1 \mid X_1, \ldots, X_{i-1}) \leq p,$$

then

$$\mathbf{Pr}(\sum Y_i \geq k) \leq \mathbf{Pr}(B(n,p) \geq k)$$

and similarly if

$$\mathbf{Pr}(Y_i = 1 \mid X_1, \ldots, X_{i-1}) \geq p,$$

then

$$\mathbf{Pr}(\sum Y_i \leq k) \leq \mathbf{Pr}(B(n,p) \leq k)$$

□

We now turn to the analysis of the finite process. In what follows, we omit the argument t, when $t = m$, that is, when the process terminates. In the interest of a clearer exposition, we start with the case $m = n$, although the general case (Theorem 8) subsumes it.

Theorem 4 *The maximum load achieved by* GREEDY *on a random (n,n,d)-problem is less than $\ln\ln n/\ln d + O(1)$ with high probability.*

Proof: Since the d choices for a ball are independent, we have

$$\mathbf{Pr}(h_t \geq i+1 \mid \nu_{\geq i}(t-1)) = \frac{\left(\nu_{\geq i}(t-1)\right)^d}{n^d}.$$

Let $\mathcal{E}_i$ be the event that $\nu_{\geq i}(n) \leq \beta_i$ where β_i will be exposed later. (Clearly, $\mathcal{E}_i$ implies that $\nu_{\geq i}(t) \leq \beta_i$ for $t = 1, \ldots, n$.) Now fix $i \geq 1$ and consider a series of binary random variables Y_t for $t = 2, \ldots, n$, where Y_t is 1 if the height of ball t is...... I.e.

$$Y_t = 1 \textbf{ iff } h_t \geq i+1 \textbf{ and } \nu_{\geq i}(t-1) \leq \beta_i.$$

(Y_t is 1 if the height of the ball t is $\geq i+1$ despite the fact that the number of boxes that have load $\geq i$ is less than β_i.)

Let ω_j represent the choices available to the j'th ball. Clearly

$$\mathbf{Pr}(Y_t = 1 \mid \omega_1, \ldots, \omega_{t-1}) \leq \frac{\beta_i^d}{n^d} \stackrel{def}{=} p_i.$$

Thus we can apply Lemma 3 to conclude that

$$\mathbf{Pr}(\sum Y_t \geq k) \leq \mathbf{Pr}(B(n, p_i) \geq k). \tag{1}$$

Observe that conditioned on $\mathcal{E}_i$, we have $\mu_{\geq i+1} = \sum Y_t$. Therefore

$$\begin{aligned} \mathbf{Pr}(\mu_{\geq i+1} \geq k \mid \mathcal{E}_i) &= \mathbf{Pr}(\sum Y_t \geq k \mid \mathcal{E}_i) \\ &\leq \frac{\mathbf{Pr}(\sum Y_t \geq k)}{\mathbf{Pr}(\mathcal{E}_i)}. \end{aligned} \tag{2}$$

Combining equations (1) and (2) we obtain that

$$\begin{aligned} \mathbf{Pr}(\nu_{\geq i+1} \geq k \mid \mathcal{E}_i) &\leq \mathbf{Pr}(\mu_{\geq i+1} \geq k \mid \mathcal{E}_i) \\ &\leq \frac{\mathbf{Pr}(B(n, p_i) \geq k)}{\mathbf{Pr}(\mathcal{E}_i)}. \end{aligned} \tag{3}$$

We can bound large deviations in the binomial distribution with the formula (see for instance [1], Appendix A.)

$$\mathbf{Pr}(B(n, p_i) \geq e p_i n) \leq e^{-p_i n}, \tag{4}$$

which inspires us to set

$$\beta_{i+1} = \frac{e\beta_i^d}{n^{d-1}}, \qquad \text{and} \qquad \beta_6 = \frac{n}{2e}.$$

With these choices $\mathcal{E}_{\geq 6} = \nu_6 \leq n/(2e)$ holds with certainty, and from 3 and 4

$$\mathbf{Pr}(\neg\mathcal{E}_{i+1} \mid \mathcal{E}_i) \leq \frac{1}{n^2 \mathbf{Pr}(\mathcal{E}_i)},$$

provided that $p_i n \geq 2\ln n$. To finish the proof let i^* be the smallest i such that $\beta_{i^*}^d / n^d \leq 2\ln n/n$. Notice that $i^* \leq \ln\ln n/\ln d + O(1)$ since

$$\beta_{i+6} = \frac{n e^{(d^i - 1)/(d-1)}}{(2e)^{d^i}} \leq \frac{n}{2^{d^i}}.$$

As before

$$\begin{aligned} &\mathbf{Pr}(\nu_{\geq i^*+1} \geq 6\ln n \mid \mathcal{E}_{i^*}) \\ &\qquad \leq \frac{\mathbf{Pr}(B(n, 2\ln n/n) \geq 6\ln n)}{\mathbf{Pr}(\mathcal{E}_{i^*})} \leq \frac{1}{n^2 \mathbf{Pr}(\mathcal{E}_{i^*})}, \end{aligned}$$

and finally

$$\begin{aligned} &\mathbf{Pr}(\mu_{\geq i^*+2} \geq 1 \mid \nu_{\geq i^*+1} \leq 6\ln n) \\ &\qquad \leq \frac{\mathbf{Pr}(B(n, (6\ln n/n)^d) \geq 1)}{\mathbf{Pr}(\nu_{\geq i^*+1} \leq 6\ln n)} \\ &\qquad \leq \frac{n(6\ln n/n)^d}{\mathbf{Pr}(\nu_{\geq i^*+1} \leq 6\ln n)} \end{aligned}$$

by Markov inequality.

We remove the conditioning using the fact that

$$\mathbf{Pr}(\neg\mathcal{E}_{i+1}) \leq \mathbf{Pr}(\neg\mathcal{E}_{i+1}) \mid \mathcal{E}_i)\mathbf{Pr}(\mathcal{E}_i) + \mathbf{Pr}(\neg\mathcal{E}_i),$$

and obtain that

$$\mathbf{Pr}(\mu_{\geq i^*+2} \geq 1) \leq \frac{(6\ln n)^d}{n^{d-1}} + \frac{i^*+1}{n^2} = o(1),$$

which implies that with high probability the maximum load achieved by GREEDY is less than $i^* + 2 = \ln\ln n/\ln d + O(1)$. □

We now prove a matching lower bound.

Theorem 5 *The maximum load achieved by* GREEDY *on a random (n, n, d)-problem is at least $\ln\ln n/\ln d - O(1)$ with high probability.*

Proof: Let $\mathcal{F}_i$ be the event that $\nu_{\geq i}(n(1 - 1/2^i)) \geq \gamma_i$ where γ_i will be exposed later. For the time being, suffices to say that $\gamma_{i+1} < \gamma_i/2$. We want to compute $\mathbf{Pr}(\neg\mathcal{F}_{i+1} \mid \mathcal{F}_i)$. To this aim, for t in the range $n(1 - 1/2^i) < t \leq n(1 - 1/2^{i+1})$, let Z_t be defined by

$$Z_t = 1 \textbf{ iff } h_t = i+1 \textbf{ or } \nu_{\geq i+1}(t-1) \geq \gamma_{i+1},$$

and observe that while $\nu_{\geq i+1}(t-1) < \gamma_{i+1}$, if $Z_t = 1$ then the box where the i'th ball is placed had load exactly i at time $t-1$. This means the ball had $d-1$ choices with load $\geq i$ and one choice with load exactly i.

Let ω_j represent the choices available to the j'th ball In view of the observation above, by considering the cases $\nu_{\geq i+1}(t-1) \geq \gamma_{i+1}$ and its complement, we derive that

$$\begin{aligned} \mathbf{Pr}(Z_t = 1 \mid \omega_1, \ldots, \omega_{t-1}, \mathcal{F}_i) &\geq \frac{\gamma_i^{d-1}(\gamma_i - \gamma_{i+1})}{n^d} \\ &\geq \frac{1}{2}\left(\frac{\gamma_i}{n}\right)^d \stackrel{def}{=} p_i. \end{aligned} \tag{5}$$

Applying Lemma 3 we get

$$\mathbf{Pr}\Bigg(\sum_{t=n(1-1/2^i)}^{n(1-1/2^{i+1})} Z_t \leq k \;\Bigg|\; \mathcal{F}_i\Bigg) \leq \mathbf{Pr}(B(n/2^{i+1}, p_i) \leq k)$$

We now choose

$$\gamma_0 = n;$$
$$\gamma_{i+1} = \frac{\gamma_i^d}{2^{i+3}n^{d-1}} = \frac{n}{2^{i+3}}\left(\frac{\gamma_i}{n}\right)^d = \frac{1}{2}\frac{n}{2^{i+1}}p_i.$$

Since $\mathbf{Pr}(B(N,p) < Np/2) < e^{-Np/8}$, it follows that

$$\mathbf{Pr}(B(n/2^{i+1}, p_i) \le \gamma_{i+1}) = o(1/n^2) \qquad (6)$$

provided that $p_i n/2^{i+1} \ge 17 \ln n$. Let i^* be the largest integer for which this holds. Clearly $i^* = \ln\ln n/\ln d - O(1)$.

Now observe that by the definition of $\mathcal{F}$ and Z_t, and in view of (5) and (6)

$$\mathbf{Pr}(\neg\mathcal{F}_{i+1} \mid \mathcal{F}_i) \le \mathbf{Pr}\left(\sum Z_t \le \gamma_{i+1} \;\middle|\; \mathcal{F}_i\right) = o(1/n^2),$$

and therefore

$$\begin{aligned}\mathbf{Pr}(\mathcal{F}_{i^*}) &\ge \mathbf{Pr}(\mathcal{F}_{i^*} \mid \mathcal{F}_{i^*-1}) \times \mathbf{Pr}(\mathcal{F}_{i^*-1} \mid \mathcal{F}_{i^*-2}) \\ &\quad \times \cdots \times \mathbf{Pr}(\mathcal{F}_1 \mid \mathcal{F}_0) \times \mathbf{Pr}(\mathcal{F}_0) \\ &\ge (1 - 1/n^2)^{i^*} = 1 - o(1/n)\end{aligned}$$

which completes the proof. □

We now turn to showing that the greedy algorithm is stochastically optimal. We say that a vector $\bar{v} = (v_1, v_2, \ldots, v_n)$ *majorizes* a vector $\bar{u}$, written $\bar{v} \succeq \bar{u}$ if for $1 \le i \le n$, we have $\sum_{1\le j\le i} v_{\pi(j)} \ge \sum_{1\le j\le i} u_{\sigma(j)}$, where π and σ are permutations of $1, \ldots, n$ such that $v_{\pi(1)} \ge v_{\pi(2)} \ge \cdots \ge v_{\pi(n)}$ and $u_{\sigma(1)} \ge u_{\sigma(2)} \ge \cdots \ge u_{\sigma(n)}$.

The proof of the following lemma is left to the full paper.

Lemma 6 *Let $\bar{v}$ and $\bar{u}$ be two positive integer vectors such that $v_1 \ge v_2 \ge \cdots \ge v_n$ and $u_1 \ge u_2 \ge \cdots \ge u_n$. If $\bar{v} \succeq \bar{u}$ then also $\bar{v} + \bar{e}_i \succeq \bar{u} + \bar{e}_i$, where $\bar{e}_i$ is the i'th unit vector, that is $\bar{e}_{i,j} = \delta_{i,j}$.*

Let Ω be the set of all possible n^d choices for each ball and Ω^t be the set of sequences of choices for the first t balls.

Theorem 7 *For any deterministic algorithm A, and $t \ge 0$, there is 1-1 correspondence $f : \Omega^t \to \Omega^t$ such that for any $\omega_t \in \Omega^t$ the vector of box loads associated to* GREEDY *acting on ω_t, written $\bar{\lambda}^G(\omega_t) = (\lambda_1^G(\omega_t), \lambda_2^G(\omega_t), \ldots, \lambda_n^G(\omega_t))$ is majorized by the vector of box counts associated to A acting on $f(\omega_t)$, that is*

$$\bar{\lambda}^G(\omega_t) \preceq \bar{\lambda}^A(f(\omega_t)).$$

Proof: To simplify notation we assume $d = 2$. The proof for larger d is analogous. The proof proceeds by induction on t, the length of the sequence. The base case is obvious. Assume the theorem valid for t and let f_t be the mapping on Ω^t. Fix a sequence $\omega_t \in \Omega^t$. Suffices to show that we can refine f_t to obtain a 1-1 correspondence for all possible 1-step extensions of ω_t. Without loss of generality, renumber the boxes such that

$$\lambda_1^G(\omega_t) \ge \lambda_2^G(\omega_t) \ge \cdots \ge \lambda_n^G(\omega_t),$$

and let π be a permutation of $1, \ldots, n$ such that

$$\lambda_{\pi(1)}^A(f_t(\omega_t)) \ge \lambda_{\pi(2)}^A(f_t(\omega_t)) \ge \cdots \ge \lambda_{\pi(n)}^A(f_t(\omega_t)).$$

Let (i,j) be two choices for the $t+1$ ball. For every i,j we define

$$f_{t+1}(\omega_t \diamond (i,j)) = f_t(\omega_t) \diamond (\pi(i), \pi(j))$$

where "$\diamond$" represents extension of sequences.

Clearly f_{t+1} is 1-1. We need to show that

$$\bar{\lambda}^G(\omega_t \diamond (i,j)) \preceq \bar{\lambda}^A(f_t(\omega_t) \diamond (\pi(i), \pi(j))).$$

Notice that when the sequence ω_t is extended by the step (i,j), for any algorithm, exactly one component of the vector $\bar{\lambda}(\omega_t)$ changes, namely either $\lambda_i(\omega_t)$ or $\lambda_j(\omega_t)$ increases by one. Assume that $i \ge j$; then

$$\begin{aligned}\bar{\lambda}^G(\omega_t \diamond (i,j)) &= \bar{\lambda}^G(\omega_t) + \bar{e}_i \\ &\preceq \bar{\lambda}^A(f_t(\omega_t)) + \bar{e}_{\pi(i)} \\ &\preceq \bar{\lambda}^A\Big(f_t(\omega_t) \diamond (\pi(i), \pi(j))\Big),\end{aligned}$$

where the first inequality follows from the Lemma 6 and the second is due to the fact that

$$\bar{\lambda}^A(f_t(\omega_t)) + \bar{e}_{\pi(i)} \preceq \bar{\lambda}^A(f_t(\omega_t)) + \bar{e}_{\pi(j)}.$$

□

We are now ready to discuss the general case of the finite process.

Theorem 8 *The maximum load achieved by* GREEDY *on a random (m,n,d)-problem, $d \ge 2$ and $m \ge n$, is less than $\ln\ln n/\ln d(1 + o(1)) + O(m/n)$ with high probability.*

Proof: We start by replaying the proof of Theorem 4, taking into account the fact that there are now m balls. So let $\mathcal{E}_i$ be the event that $\nu_{\ge i}(m) \le \beta_i$, and define $p_i = \beta_i^d/n^d$. Following the proof of Theorem 4 we derive that

$$\mathbf{Pr}(\nu_{\ge i+1} \ge k \mid \mathcal{E}_i) \le \frac{\mathbf{Pr}(B(m,p_i) \ge k)}{\mathbf{Pr}(\mathcal{E}_i)}.$$

Suppose that for some value x we set $\beta_x = n^2/(2em)$ and show that $\mathcal{E}_x$ holds with high probability, that is

$$\mathbf{Pr}\left(\nu_x \ge \frac{n^2}{2em}\right) = o(1). \qquad (7)$$

Then

$$\beta_{i+x} = \frac{n}{2^{d^i}} \left(\frac{me}{n}\right)^{(d^i-1)/(d-1)-d^i} \leq \frac{n}{2^{d^i}},$$

and continuing as before, we obtain that

$$\mathbf{Pr}(\mu \geq x + \ln\ln n/\ln d + 2) = o(1).$$

It remains to show that x can be taken to be $O(m/n) + o(\ln\ln n/\ln d)$. First assume that $m/n \geq w(n)$ where $w(n)$ is an increasing function of n, but $w(n) = o(\ln\ln n/\ln d)$. Then we claim that we can take $x = \lceil em/n \rceil$.

Consider a placement algorithm, denoted R, that always puts a ball in the box corresponding to the first choice offered. This is entirely equivalent with the case $d = 1$, the classical occupancy problem. The load within a box under this process is a binomial random variable $B(m, 1/n)$, and therefore (via (4)), we obtain that the expected number of boxes with load $\geq em/n$ satisfies

$$\mathbf{E}(\nu^R_{\geq em/n}) \leq ne^{-m/n}.$$

Hence by Markov's inequality

$$\mathbf{Pr}\left(\nu^R_{\geq em/n} \geq \frac{n^2}{2em}\right) \leq \frac{2m}{n} e^{-m/n+1} = o(1),$$

since $m/n \to \infty$. Now we can apply Theorem 7 to show that $x = \lceil em/n \rceil$ satisfies equation (7).

To remove the assumption $m/n \geq w(n)$, we can simply imagine that the number of balls is increased to $\max(m, nw(n))$. Then the corresponding value of x becomes $O(\max(m, nw(n))/n) = O(m/n) + o(\ln\ln n/\ln d)$. □

4 The Infinite Process

In this section we consider the infinite process. Analogously to Theorem 7 it is possible to show that the greedy algorithm minimizes the expected maximum load on any box. We analyse its performance below. The main theorem of this section is

Theorem 9 *Assume that the infinite process starts in an arbitrary state. Under* GREEDY, *with* $d \geq 2$, *for any fixed* $T \geq n^3$,

$$\mathbf{Pr}(\exists j, \lambda_j(T) \geq \ln\ln n/\ln d + O(1)) = o(1).$$

Thus in the stationary distribution the maximum load is $\ln\ln n/\ln d + O(1)$ *with high probability.*

Proof: For simplicity of presentation we state and prove the results only for $d = 2$. The proof assumes that at time $T - n^3$ the process is in an arbitrary state and therefore we can let $T = n^3$ with no loss of generality.

By the definition of the process, the number of balls of height at least i cannot change by more than 1 in a time step, that is $|\mu_{\geq i}(t+1) - \mu_{\geq i}(t)| \leq 1$. The random variable $\mu_{\geq i}(t)$ can be viewed as a random walk on the integers l, $0 \leq l \leq n$. The proof is based on bounding the maximum values taken by the variables $\mu_{\geq i}(t)$ by studying the underlying process.

We define an integer i^* and a decreasing sequence α_i, for $40 \leq i \leq i^* + 1$ as follows:

$$\begin{aligned}
\alpha_{40} &= \frac{n}{40}, \\
\alpha_i &= \frac{20\alpha_{i-1}^2}{n}, && \text{for } i > 40 \text{ and } \alpha_{i-1} \geq \sqrt{12n\log_2 n/20}, \\
\alpha_{i^*} &= 12\log_2 n, && i^* = \text{the smallest } i \text{ for which } \alpha_{i-1} < \sqrt{12n\log_2 n/20}, \\
\alpha_{i^*+1} &= 16.
\end{aligned}$$

Clearly $i^* \leq \ln\ln n/\ln 2 + O(1)$.

We also define an increasing sequence of times: $t_{40} = 0$ and $t_i = t_{i-1} + n^2$ for $i > 40$. Thus $t_{i^*+1} = O(n^2 \log\log n) = o(T)$.

Let $\{\mu_{\geq i}[t^-, t^+] \leq \alpha\}$ denote the event that $\mu_{\geq i}(t) \leq \alpha$ for all t, such that $t^- \leq t \leq t^+$, and similarly, let $\{\mu_{\geq i}[t^-, t^+] > \alpha\}$ denote the event that $\mu_{\geq i}(t) > \alpha$ for all t, such that $t^- \leq t \leq t^+$. We define the events C_i as follows:

$$\begin{aligned}
C_{40} &= \{\nu_{\geq 40}[t_{40}, T] \leq 2\alpha_{40}\} \equiv \{\nu_{\geq 40}[0, T] \leq n/20\}; \\
C_i &= \{\mu_{\geq i}[t_i, T] \leq 2\alpha_i\}, \qquad \text{for } i > 40.
\end{aligned}$$

We shall prove inductively that for all $i = 40, \ldots, i^* + 1$

$$\mathbf{Pr}(\neg C_i) \leq \frac{3i}{n^6}. \tag{8}$$

This implies that the event $\{\mu_{\geq i^*+1}[t_{i^*+1}, T] \leq 32\}$ occurs with probability $1 - o(1)$, and therefore with high probability, for every j, $\lambda_j(T) \leq i^* + 33 = \ln\ln n/\ln 2 + O(1)$ which completes the proof of the Theorem.

We now return to the proof of (8), which is done by proving that conditioned on C_{i-1}

- With high probability $\mu_{\geq i}(t)$ becomes less than α_i before time t_i. (Lemma 10.)
- If $\mu_{\geq i}(t)$ becomes less than α_i at any time before T, then from then on until T, with high probability, it does not become larger than $2\alpha_i$. (Lemma 11.)

The two facts above imply that if C_{i-1} holds, then with high probability $\mu_{\geq i}[t_i, T] \leq 2\alpha_i$, that is C_i holds as well.

Base Case: The base case is trivial since $\mathbf{Pr}(\neg C_{40}) = \mathbf{Pr}(\neg\{\nu_{\geq 40}[0, T] \leq n/20\}) = 0$.

Induction: Suppose that

$$\mathbf{Pr}(\neg C_{i-1}) \le \frac{3(i-1)}{n^6}, \tag{9}$$

where $40 < i \le i^* + 1$.

Let S_i^l be the set of states such that $\mu_{\ge i} = l$ and let $s(t)$ be the state at time t. It is easy to verify the following bounds on the underlying transition probabilities. For any t and any $i > 2$,

$$\mathbf{Pr}(\mu_{\ge i}(t+1) > \mu_{\ge i}(t) \mid s(t)) \le \left(\frac{\nu_{\ge(i-1)}(t)}{n}\right)^2 \le \left(\frac{\mu_{\ge(i-1)}(t)}{n}\right)^2 \tag{10}$$

and

$$\mathbf{Pr}(\mu_{\ge i}(t+1) < \mu_{\ge i}(t) \mid s(t)) \ge \frac{\mu_{\ge i}(t)}{n}\left(1 - \left(\frac{\nu_{\ge(i-1)}(t)}{n}\right)^2\right) \ge \frac{\mu_{\ge i}(t)}{2n} \tag{11}$$

Two key lemmas are necessary in order to conclude that $\mathbf{Pr}(\neg C_i) \le 3i/n^6$.

Lemma 10 *Under the inductive hypothesis*

$$\mathbf{Pr}(\mu_{\ge i}[t_{i-1}, t_i] > \alpha_i \mid C_{i-1}) \le \frac{1}{n^6}.$$

Proof: From equations (10) and (11) we obtain that the transition probabilities satisfy

$$\mathbf{Pr}(\mu_{\ge i}(t+1) > \mu_{\ge i}(t) \mid s(t), \mu_{\ge i-1}(t) \le 2\alpha_{i-1}) \le \left(\frac{2\alpha_{i-1}}{n}\right)^2 \stackrel{def}{=} q_i^+,$$

and

$$\mathbf{Pr}(\mu_{\ge i}(t+1) < \mu_{\ge i}(t) \mid s(t), \mu_{\ge i}(t) \ge \alpha_i) \ge \frac{\alpha_i}{2n} \stackrel{def}{=} q_i^-.$$

We define two new binary random variables for $40 < t \le T$ as follows:

$$X_t = 1 \textbf{ iff } \mu_{\ge i}(t) > \mu_{\ge i}(t-1) \textbf{ and } \mu_{\ge i-1}(t-1) \le 2\alpha_{i-1},$$

and

$$Y_t = 1 \textbf{ iff } \mu_{\ge i}(t) < \mu_{\ge i}(t-1) \textbf{ or } \mu_{\ge i}(t-1) \le \alpha_i.$$

Clearly

$$\mathbf{Pr}(X_t = 1) \le q_i^+ \qquad \text{and} \qquad \mathbf{Pr}(Y_t = 1) \ge q_i^- \tag{12}$$

Notice that conditioned on C_{i-1}, the sum $\sum_{t\in[t_{i-1},t_i]} X_t$ is the number of times $\mu_{\ge i}(t)$ increased in the interval $[t_{i-1}, t_i]$; similarly, if within this interval $\mu_{\ge i}$ did not become less than α_i, then $\sum_{t\in[t_{i-1},t_i]} Y_t$ equals the number of times $\mu_{\ge i}(t)$ decreased in this interval. We conclude that

$$\begin{aligned}\mathbf{Pr}&(\mu_{\ge i}[t_{i-1}, t_i] > \alpha_i \mid C_{i-1}) \\ &\le \mathbf{Pr}\left(\sum_{t\in[t_{i-1},t_i]} Y_t - \sum_{t\in[t_{i-1},t_i]} X_t \le n \;\middle|\; C_{i-1}\right) \\ &\le \frac{1}{\mathbf{Pr}(C_{i-1})}\mathbf{Pr}\left(\sum_{t\in[t_{i-1},t_i]} Y_t - \sum_{t\in[t_{i-1},t_i]} X_t \le n\right)\end{aligned}$$

In view of equation (12) and Lemma 3, Chernoff type bounds imply that for every $i \le i^* + 1$

$$\begin{aligned}\mathbf{Pr}\left(\sum_{t\in[t_{i-1},t_i]} X_t > \tfrac{5}{4}n^2 q_i^+\right) &\le \mathbf{Pr}\left(B(n^2, q_i^+) \ge \tfrac{5}{4}n^2 q_i^+\right) \\ &\le e^{-\Omega(n^2 q_i^+)} = o(1/n^c),\end{aligned}$$

and

$$\begin{aligned}\mathbf{Pr}\left(\sum_{t\in[t_{i-1},t_i]} Y_t < \tfrac{3}{4}n^2 q_i^-\right) &\le \mathbf{Pr}(B(n^2, q_i^-) \le \tfrac{3}{4}n^2 q_i^-) \\ &\le e^{-\Omega(n^2 q_i^-)} = o(1/n^c),\end{aligned}$$

for any constant c. On the other hand

$$\frac{3}{4}n^2 q_i^- - \frac{5}{4}n^2 q_i^+ \ge \frac{3}{8}n\alpha_i - 5\alpha_{i-1}^2 \ge \frac{n\alpha_i}{8} \ge n,$$

and therefore we conclude that

$$\mathbf{Pr}(\mu_{\ge i}[t_{i-1}, t_i] > \alpha_i \mid C_{i-1}) \le \frac{1}{n^c \mathbf{Pr}(C_{i-1})},$$

for any constant c. Taking $c = 13$ and using the inductive hypothesis on C_{i-1} (equation (9)) yields the theorem. $\square$

Lemma 11 *Under the inductive hypothesis, for any t^* between t_{i-1} and t_i*

$$\mathbf{Pr}(\neg C_i \mid C_{i-1}, \{\mu_{\ge i}(t^*) \le \alpha_i\}) \le 2/n^6.$$

Proof: Define the event $D_i = D_i(t^*)$ by

$$D_i(t^*) = C_{i-1} \textbf{ and } \{\mu_{\ge i}(t^*) \le \alpha_i\}.$$

Let $X_t(l)$ be an indicator random variable which is 1 if $s(t) \in S_i^l$ and 0 otherwise. Then $\sum_{t^* \le t \le T} X_t(l)$ is the number of times $s(t) \in S_i^l$ between t^* and T. Let $\Xi_i(l)$ be the expectation of this sum given the event D_i, i.e.

$$\Xi_i(l) = \mathbf{E}\left(\sum_{t^* \le t \le T} X_t(l) \;\middle|\; D_i\right).$$

The transition probabilities and equation 9 imply that for $t^* \le t \le T$

$$\begin{aligned}
&\mathbf{Pr}(X_{t+1}(l+1) = 1 \mid X_t(l) = 1, D_i) \\
&\quad \le \mathbf{Pr}(X_{t+1}(l+1) = 1 \mid \\
&\qquad\qquad X_t(l) = 1, \mu_{\ge i}(t^*) \le \alpha_i, \mu_{\ge i-1}(t) \le 2\alpha_{i-1}) \\
&\qquad \times \frac{1}{\mathbf{Pr}(C_{i-1})} \\
&\quad \le \left(\frac{2\alpha_{i-1}}{n}\right)^2 \frac{1}{\mathbf{Pr}(C_{i-1})} \le \frac{5}{4}\left(\frac{2\alpha_{i-1}}{n}\right)^2 \stackrel{def}{=} q_i^+
\end{aligned}$$

and for $l \ge \alpha_i$

$$\begin{aligned}
&\mathbf{Pr}(X_{t+1}(l) = 1 \mid X_t(l+1) = 1, D_i) \\
&\quad \ge \mathbf{Pr}(X_{t+1}(l) = 1 \mid X_t(l+1) = 1, \mu_{\ge i}(t^*) \le \alpha_i) \\
&\qquad - \mathbf{Pr}(\neg C_{i-1}) \\
&\quad \ge \frac{l+1}{2n} - \frac{1}{2n} \ge \frac{\alpha_i}{2n} \stackrel{def}{=} q_i^-
\end{aligned}$$

For a random walk on the line starting at a point x, the number of transitions from $y \ge x$ to $y+1$ is either the same or one greater than the number of transitions from $y+1$ to y. Thus we claim that for any $l \ge \alpha_i$

$$\begin{aligned}
&\mathbf{E}\left(\sum_{t^* \le t \le T} X_{t+1}(l+1)X_t(l) \;\middle|\; D_i\right) \\
&\qquad \ge \mathbf{E}\left(\sum_{t^* \le t \le T} X_t(l+1)X_{t+1}(l) \;\middle|\; D_i\right).
\end{aligned} \tag{13}$$

Indeed, D_i implies that at time t^*, $s(t) \in S_i^l$, for some $l \le \alpha_i$. Hence for $l \ge \alpha_i$, the total number of transitions from l to $l+1$ is at least the total number of transitions from $l+1$ to l. Thus we have

$$\begin{aligned}
&\mathbf{E}(X_{t+1}(l+1)X_t(l) \mid D_i) \\
&\quad = \mathbf{Pr}(X_{t+1}(l+1) = 1 \mid X_t(l) = 1, D_i) \\
&\qquad \times \mathbf{Pr}(X_t(l) = 1 \mid D_i) \\
&\quad \le q_i^+ \mathbf{Pr}(X_t(l) = 1 \mid D_i),
\end{aligned}$$

and so

$$\begin{aligned}
&\mathbf{E}\left(\sum_{t^* \le t \le T} X_{t+1}(l+1)X_t(l) \;\middle|\; D_i\right) \\
&\qquad \le q_i^+ \sum_{t^* \le t \le T} \mathbf{E}(X_t(l) \mid D_i) = q_i^+ \Xi_i(l).
\end{aligned}$$

Similarly,

$$\mathbf{E}\left(\sum_{t^* \le t \le T} X_t(l+1)X_{t+1}(l) \;\middle|\; D_i\right) \ge q_i^- \Xi_i(l+1).$$

From equation (13), we have $q_i^+ \Xi_i(l) \ge q_i^- \Xi_i(l+1)$, and therefore

$$\frac{\Xi_i(l+1)}{\Xi_i(l)} \le \frac{q_i^+}{q_i^-} \le \frac{5}{4}\frac{(2\alpha_{i-1}/n)^2}{(\alpha_i/2n)} \le \frac{1}{2}.$$

Thus,

$$\Xi_i(2\alpha_i) \le \frac{\Xi_i(\alpha_i)}{2^{\alpha_i}} \le \frac{T}{2^{\alpha_i}}.$$

If $i \le i^*$, then $\alpha_i \ge 12\log_2 n$, thus $\Xi_i(2\alpha_i) \le 1/n^6$ and $\sum_{l \ge 2\alpha_i} \Xi_i(l) \le 2/n^6$. For $i = i^*+1$,

$$\left(\frac{q_{i^*+1}^+}{q_{i^*+1}^-}\right)^{\alpha_{i^*+1}} \le \left(\frac{720(\log_2 n/n)^2}{8/n}\right)^{16} \le \frac{1}{n^6}.$$

Therefore $\sum_{l \ge 2\alpha_{i^*+1}} \Xi_{i^*+1}(l) \le 2/n^6$.

Finally, the fact that

$$\mathbf{Pr}(\neg C_i \mid D_i) \le \sum_{l \ge 2\alpha_i} \Xi_i(l)$$

completes the proof. □

Returning to the proof of equation (8), let again $D_i(t^*)$ be the event C_{i-1} **and** $\{\mu_{\ge i}(t^*) \le \alpha_i\}$. Using the induction hypothesis, lemmas 10, 11 and the law of total probability, we can complete the induction, as follows:

$$\begin{aligned}
\mathbf{Pr}(\neg C_i) &= \mathbf{Pr}(\neg C_i \mid C_{i-1}) \cdot \mathbf{Pr}(C_{i-1}) \\
&\quad + \mathbf{Pr}(\neg C_i \mid \neg C_{i-1}) \cdot \mathbf{Pr}(\neg C_{i-1}) \\
&\le \mathbf{Pr}(\neg C_i \mid C_{i-1}) + 3(i-1)/n^6 \\
&= \mathbf{Pr}(\neg C_i \mid C_{i-1}, \mu_{\ge i}[t_{i-1}, t_i] > \alpha_i) \\
&\qquad \times \mathbf{Pr}(\mu_{\ge i}[t_{i-1}, t_i] > \alpha_i \mid C_{i-1}) \\
&\quad + \mathbf{Pr}(\neg C_i \mid \exists t^* \in [t_{i-1}, t_i] : D_i(t^*)) \\
&\qquad \times \mathbf{Pr}(\exists t^* \in [t_{i-1}, t_i] : D_i(t^*) \mid C_{i-1}) \\
&\quad + 3(i-1)/n^6 \\
&\le 1/n^6 + 2/n^6 + 3(i-1)/n^6 = 3i/n^6.
\end{aligned}$$

□

5 Competitive Online Load Balancing

5.1 Preliminaries. We define the online load balancing problem precisely. Let M be a set of servers (or machines) that is supposed to run a set of tasks that arrive and depart in time. Each task j has associated to it a weight, or load, $w(j) \ge 0$, an arrival time $\tau(j)$, and a set $M(j) \subset M$ of servers capable of running it.

As soon as it arrives, each task must be assigned to exactly one of the servers capable of running it, and once assigned, it can not be transferred to a different server. The assigned server starts to run the task immediately, and continues to run it until the task departs.

When task j arrives, an *assignment algorithm* must select a server $i \in M(j)$, and assign task j to it.

The *load* on server i at time t, denoted $L_i^A(t)$, is the sum of the weights of all the tasks running on server i at time t when assignment algorithm A is used. When A is obvious, the superscript is omitted.

Let σ be a sequence of task arrivals and departures, and let $|\sigma|$ be the time of the last arrival. Then the cost, $C_A(\sigma)$, of an assignment algorithm A on the sequence σ, is defined as

$$C_A(\sigma) = \max_{0 \le t \le |\sigma|; i \in M} L_i(t).$$

An *on-line assignment algorithm* must assign an arriving task j at time $\tau(j)$ to a server in $M(j)$ knowing only $w(j)$, $M(j)$, the current state of the servers, and the past – the decision is made without any knowledge about future arrivals or departures. An *optimal off-line assignment algorithm*, denoted OPT, assigns arriving tasks knowing the entire sequence of task arrivals and departures and does so in a way that minimizes its cost.

The competitive ratio of an on-line algorithm A is defined as the supremum over all sequences σ of the ratio $C_A(\sigma)/C_{\mathrm{OPT}}(\sigma)$.

Let $C_A(\mathcal{P})$ (resp. $C_{\mathrm{OPT}}(\mathcal{P})$) be the expected cost of algorithm A (resp.OPT) on sequences σ generated by the distribution $\mathcal{P}$. The competitive ratio of an on-line algorithm A *against distribution* $\mathcal{P}$ is defined as the ratio $C_A(\mathcal{P})/C_{\mathrm{OPT}}(\mathcal{P})$.

Finally, the greedy algorithm is formally defined as follows:

Algorithm GREEDY**:** Upon arrival of a task j assign it to the server in $M(j)$ with the current minimum load (ties are broken arbitrarily).

5.1.1 Permanent Tasks.

For permanent tasks, Azar, Naor and Rom [7] showed that the competitive ratio of the greedy algorithm is $\Theta(\log n)$ and that no algorithm can do better.

This large competitive ratio hinges on the fact that an adversary can construct a particularly malicious sequence of task types. In other words, it is a worst case result. It is interesting to consider the average case. Hence we consider input sequences generated from a distribution, and show that the competitive ratio against such distributions is exponentially smaller than against a worst-case adversary.

For simplicity in this abstract, we present our results for the case where for each task j, $M(j)$ is a random subset of M of cardinality $d \geq 2$ (chosen with replacement), $|\sigma| = n$, and all weights are equal. Let $\mathcal{P}_d$ be the associated probability distribution on request sequences. It is possible to extend some of these results to the case where machine $i \in M$ is in $M(j)$ with probability p_i, the sequences are of arbitrary length and the weights are arbitrary.

We omit the proof of the following bound on the performance of the optimal offline algorithm.

Lemma 12 *With probability* $1 - O(1/n)$, $C_{\mathrm{OPT}}(\mathcal{P}_d) = O(1)$.

Lemma 13 *With high probability,* $C_{\mathrm{GREEDY}}(\mathcal{P}_d) = O(\log\log n / \log d)$

Proof: Follows immediately from theorem 4. □

Thus, we obtain the following theorem.

Theorem 14 *The competitive ratio of the* GREEDY *algorithm against the distribution* $\mathcal{P}_d$ *is* $O(\log\log n/\log d)$ *and no algorithm can do better.*

Proof: Follows from lemmas 12, 13 and theorem 7. □

5.1.2 Temporary Tasks.

For temporary tasks, the works of Azar, Broder and Karlin,[5] and Azar, Kalyanasundaram, Plotkin, Pruhs and Waarts [6] showed that there is an algorithm with competitive ratio $\Theta(\sqrt{n})$ and that no algorithm can do better.

It is difficult to construct a natural distribution of task arrivals and departures. As a first step, we consider the following stochastic process $\mathcal{S}$: First, n tasks arrive, where each may be served by d servers, chosen uniformly and independently at random (with replacement). Then forever the following repeats: a random task among those present departs, and a random task arrives, which again may be served by one of d random servers.

Clearly, in such an infinite sequence, there will eventually be n tasks which can only be served by one server, and so for silly reasons the competitive ratio will be 1. Therefore we state our competitiveness result in the following way:

Theorem 15 *Let* $L_A[t]$ *be the maximum load on any server at time* t*, for tasks arriving according to the stochastic process* $\mathcal{S}$*, and assigned using algorithm* A*. i.e.* $L_A[t] = \max_{i \in M} L_i^A(t)$*. Then for any fixed* $t > 0$*, with high probability,*

$$\frac{L_{\mathrm{GREEDY}}[t]}{L_{\mathrm{OPT}}[t]} = O(\log\log n).$$

Proof: Follows from lemma 12, and theorem 9. □

Acknowledgement. We wish to thank Martin Dyer and Alan Frieze for several very useful discussions.

A An alternative derivation of Theorem 4

In this section we show how it is possible to derive a weaker form of Theorem 4 for $d = 2$ following the ideas outlined by Karp, Luby, and Meyer auf der Heide in [13].

Let $U = \{1, \ldots, n\}$. Let h_1 and h_2 be two random functions $U \to U$. Let S be an arbitrary subset of U of size $n/8$. Consider the following process:

```
Repeat until S = ∅
    For j = 1, 2 do
        For i ∈ {1, ..., n} do
            Remove one x ∈ h_j^{-1}(i) from S
```

Theorem 4 of [13] states that with high probability the repeat loop is executed only $O(\log \log n)$ times. The proof is based on the analysis of sparse random graphs on n vertices: each $i \in S$ is associated to the edge $(h_1(i), h_2(i))$.

To use this theorem in our setting, view the two choices available to ball i as two random functions $h_1(i)$ and $h_2(i)$. Define an algorithm MGREEDY, that acts as follows:

When ball i arrives:

- Let $h_1(i) = x$ and $h_2(i) = y$.
- The index of i in x is the number of balls that went via h_1 to box x so far; The index of i in y is the number of balls that went via h_2 to box y so far.
- Put ball i in the box where it gets a lower index.

To get the desired bound notice that:

a. By Theorem 7, GREEDY dominates any other algorithm, in particular MGREEDY. So it is enough to show that the maximum load for MGREEDY is $O(\log \log n)$.
b. It is necessary and sufficient to show that each of the two hash functions never puts more than $O(\log \log n)$ balls in each box.
c. In the process above the order of removal in the innermost step is arbitrary. In particular we can set the order such that the balls removed in iteration k of the "repeat" loop are precisely the balls with index k.
d. The number of outer iterations is exactly the maximum index.

Thus Theorem 4 of [13] implies that MGREEDY, and and a fortiori GREEDY, has a maximum load that is $O(\log \log n)$ with high probability.

Although a direct analysis of the sparse random graph model might lead to bounds as accurate, or even better than the bound $\ln \ln n / \ln 2 + O(1)$, that we obtained in Theorem 4, it seems to be difficult to extend the analysis to the case $d > 2$ and/or to the infinite process.

References

[1] N. Alon and J. Spencer. The probabilistic method. John Wiley and Sons, 1992.

[2] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, O. Waarts. Online machine scheduling with applications to load balancing and virtual circuit routing. In *Proc. of 25th Symposium on Theory of Computing*, pp. 164–173, 1993.

[3] B. Awerbuch, Y. Azar and S. Plotkin. Throughput competitive online routing. In *Proceedings of the 34th IEEE Conference on Foundations of Computer Science*, pp. 32-40, 1993.

[4] B. Awerbuch, Y. Azar, S. Plotkin and O. Waarts. Competitive routing of virtual circuits with unknown duration. In *Proceedings of 5th ACM/SIAM Symposium on Discrete Algorithms*, pp. 321-327, 1994.

[5] Y. Azar, A. Z. Broder and A .R. Karlin. Online load balancing. In *Proceedings of the 33rd IEEE Conference on Foundations of Computer Science*, 1992.

[6] Y. Azar, B. Kalyanasundaram, S. Plotkin, K. Pruhs, O. Waarts. Online load balancing of temporary tasks. In *Proceedings of the Workshop on Algorithms and Data Structures*, pp. 119-130, August, 1993.

[7] Y. Azar, J. Naor and R. Rom. The competitiveness of online assignment. In *Proceedings of 3rd ACM/SIAM Symposium on Discrete Algorithms*, pp. 203–210, 1992.

[8] A. Broder and A. Karlin. Multi-level Adaptive Hashing. In *Proceedings of 1st ACM/SIAM Symposium on Discrete Algorithms*, January, 1990.

[9] M. Dietzfelbinger, A. Karlin, K. Mehlhorn, F. Meyer auf der Heide, H. Rohnert, R. Tarjan. Dynamic Perfect Hashing - Upper and Lower Bounds In *Proceedings of the 29th IEEE Conference on Foundations of Computer Science*, 1988.

[10] M. L. Fredman, J. Komlós, and E. Szemerédi. Storing a sparse table with O(1) worst case access time. *Journal of the ACM*, 31:538–544, 1984.

[11] Gaston H. Gonnet. Expected length of the longest probe sequence in hash code searching. *Journal of the ACM*, 28(2):289–304, April 1981.

[12] Norman L. Johnson and Samuel Kotz. *Urn Models and Their Application.* John Wiley & Sons, 1977.

[13] Richard M. Karp, Michael Luby, and Friedhelm Meyer auf der Heide. Efficient PRAM simulation on a distributed memory machine. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, 1992.

[14] R. M. Karp and U. V. Vazirani and V. V. Vazirani. An Optimal Algorithm for On-Line Bipartite Matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, 1990.

[15] Valentin F. Kolchin, Boris A. Sevastyanov, and Valdimir P. Chistyakov. *Random Allocations.* John Wiley & Sons, 1978.

[16] S.J. Phillips and J. Westbrook. Online load balancing and network flow. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, 1993.

[17] D.D. Sleator and R.E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28:202–208, February 1985.